UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/788,455	03/01/2004	Kent Bodell	C697 0007/GNM	7385

720 7590 03/19/2009

OYEN, WIGGS, GREEN & MUTALA LLP
480 - THE STATION
601 WEST CORDOVA STREET
VANCOUVER, BC V6B 1G1
CANADA

EXAMINER	
GUPTA, MUKTESH G	

ART UNIT	PAPER NUMBER
2444	

MAIL DATE	DELIVERY MODE
03/19/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/788,455	BODELL ET AL.	
	Examiner	Art Unit	
	Muktesh G. Gupta	2444	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 09 December 2008.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-20 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-20 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Acknowledgement is made to Applicants response dated 12/09/2008 to FOAM dated 08/20/2008.

Claims 1-20 were originally presented and not amended.

Claims 1-20 are presented have been examined on merits and are pending in this application.

Response to Arguments

2. *Applicant's arguments filed on 12/09/2008, regarding **Claims 1-20** have been fully considered, but they are not persuasive.*

a. Applicant argues that the applied reference does not suggest the feature of "a full-duplex packetized interconnect directly connecting a CPU of the first compute node to a network interface connected to the inter-node communication network"; and "a dedicated full-duplex packetized interconnect directly coupling the CPU to the network interface".

b. Arndt discloses as stated in par. 0030, par. 0035, par. 0042, SAN 100 is a high-bandwidth, low-latency network interconnecting nodes within the distributed computer system. A node is any component attached to one or more links of a network and forming the origin and/or destination of messages within the network. In SAN 100, a link is a full duplex channel between any two network fabric elements, such as end nodes, switches, or routers. Example suitable links include, but are not limited to, copper cables, optical cables, and printed circuit

copper traces on backplanes and printed circuit boards. Data processing system as described by Arndt uses InfiniBand hardware message passing mechanism which can be used for Input/Output devices (I/O) and Interprocess Communications (IPC) between general computing nodes. Consumer's access IB message passing hardware by posting send/receive messages to send/receive work queues on an IB Channel Adapter (CA). The send/receive work queues (WQ) are assigned to a consumer as a Queue Pair (QP). Consumers retrieve the results of these messages from a Completion Queue (CQ) through IB send and receive work completions (WC). The source CA takes care of segmenting outbound messages and sending them to the destination. The destination CA takes care of reassembling inbound messages and placing them in the memory space designated by the destination's consumer. There are two CA types: Host CA and Target CA. The Host Channel Adapter (HCA) is used by general purpose computing nodes to access the IB fabric. Consumers use IB verbs to access Host CA functions. The software that interprets verbs and directly accesses the CA is known as the Channel Interface (CI). SAN 100 handles data communications for I/O and interprocessor communications. SAN 100 supports high-bandwidth and scalability required for I/O and also supports the extremely low latency and low CPU overhead required for interprocessor communications. User clients can bypass the operating system kernel process and directly access network communication hardware, such as host channel adapters, which enable efficient message passing protocols. SAN 100 is suited

to current computing models and is a building block for new forms of I/O and computer cluster communication. Further, SAN 100 in FIG. 1 allows I/O adapter nodes to communicate among themselves or communicate with any or all of the processor nodes in distributed computer system. With an I/O adapter attached to the SAN 100, the resulting I/O adapter node has substantially the same communication capability as any host processor node in SAN 100. SAN 100 shown in FIG. 1 supports channel semantics and memory semantics. In memory semantics, a source process directly reads or writes the virtual address space of a remote node destination process. The remote destination process need only communicate the location of a buffer for data, and does not need to be involved in the transfer of any data. Thus, in memory semantics, a source process sends a data packet containing the destination buffer memory address of the destination process. In memory semantics, the destination process previously grants permission for the source process to access its memory.

c. Applicant argues that the applied reference does not suggest the feature of: "the network interface and the CPU are the only devices configured to place data on the packetized interconnect", and "the dedicated packetized full-duplex interconnect is not shared by any devices other than the CPU and the network interface".

d. Arndt discloses as stated in par. 0066-0070, For interprocessor communications, a user-mode software process transfers data through queue pairs directly from where the buffer resides in memory. In one embodiment, the

transfer through the queue pairs bypasses the operating system and consumes few host instruction cycles. Queue pairs permit zero processor-copy data transfer with no operating system kernel involvement. The zero processor-copy data transfer provides for efficient support of high-bandwidth and low-latency communication. Reliable and Unreliable connected services associate a local queue pair with one and only one remote queue pair. Connected services require a process to create a queue pair for each process that is to communicate with over the SAN fabric. Host processor node 1 includes queue pairs 4, 6 and 7, each having a send work queue and receive work queue. Host processor node 2 has a queue pair 9 and host processor node 3 has queue pairs 2 and 5. The reliable connection service of distributed computer system 500 associates a local queue pair with one and only one remote queue pair. Thus, the queue pair 4 is used to communicate with queue pair 2; queue pair 7 is used to communicate with queue pair 5; and queue pair 6 is used to communicate with queue pair 9.

e. Applicant argues that the applied reference does not suggest the feature of: "at the network interface, determining a size of the data and, based upon the size of the data, selecting among two or more protocols for transmitting the data", and "the network interface comprises a facility configured to determine a size of data to be transmitted to another compute node and, based upon the size, to select among two or more protocols..."

f. Arndt discloses as stated in par. 0101-0109, par. 0059, The client process calls an operating-system specific programming interface, which is

herein referred to as "verbs." The software code implementing verbs posts a work queue element to the given queue pair work queue. A user process, however, must communicate to verbs in a well-defined manner, and the format and protocols of data transmitted across the SAN fabric must be sufficiently specified to allow devices to interoperate in a heterogeneous vendor environment. An outgoing message is split into one or more data packets. In one embodiment, the channel adapter hardware adds a transport header and a network header to each packet. The transport header includes sequence numbers and other transport information. The network header includes routing information, such as the destination IP address and other network routing information. The link header contains the Destination Local Identifier (DLID) or other local routing information. The appropriate link header is always added to the packet. The appropriate global network header is added to a given packet it the destination end node resides on a remote subnet. Host channel adapter end node protocol layers (employed by end node 1111, for instance) include an upper level protocol 1102 defined by consumer 1103, a transport layer 1104; a network layer 1106, a link layer 1108, and a physical layer 1110. With respect to the protocol layers of end node 1111, for example, upper layer protocol 1102 employs verbs to create messages at transport layer 1104. Consumers 1103 and 1105 represent applications or processes that employ the other layers for communicating between end nodes. Transport layer 1104 provides end-to-end message movement. In one embodiment, the transport layer provides four types

of transport services which are reliable connection service; reliable datagram service; unreliable datagram service; and raw datagram service. Network layer 1106 performs packet routing through a subnet or multiple subnets to destination end nodes. Link layer 1108 performs flow-controlled, error checked, and prioritized packet delivery across links. A completion queue element is a data structure on a completion queue. This element describes a completed work queue element. The completion queue element contains sufficient information to determine the queue pair and specific work queue element that completed. A completion queue context is a block of information that contains pointers to, length, and other information needed to manage the individual completion queues.

Therefore, Applicant's arguments are not persuasive regarding all these features, elements of the present application.

Hence Examiner respectfully disagrees with Applicant's arguments on page 2-6, and maintains his rejection.

Applicant had opportunity to amend the claimed subject matter, and has failed to modify the claim language to distinguish over the prior art of record by clarifying or substantially narrowing the claim language. Thus, Applicant apparently intends that a broad interpretation be given to the claims and the Examiner has adopted such in the present and previous Office action rejections. See In re Prater and Wei, 162 USPQ 541 (CCPA 1969), and MPEP 2111.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. **Claims 1-20** rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Application Publication No. 20050018669 to Arndt, Richard Louis et al., (hereinafter "Arndt") as applied to **Claims 1-20** above, and further in view of U.S. Patent No. 6542513 to Franke; Hubertus et al., (hereinafter "Franke").

Regarding **Claims 6-7 and 13-14,** Arndt disclosed the invention substantially as claimed regarding using two or more protocols for transmitting the data. Arndt does not explicitly disclose "two or more protocols comprise an eager protocol and a rendezvous protocol".

Arndt shows, for Interprocessor communications; a user-mode software process transfer's data through queue pairs directly from where the buffer resides in memory. When a queue pair is created, the queue pair is set to provide a selected type of transport service. Distributed computer system supports four types of transport services: reliable connection, unreliable connection, reliable datagram, and unreliable datagram connection service. Arndt does not show protocols comprise an eager protocol and a rendezvous protocol.

Franke shows in a message processing system having message source and destination nodes, FIG. 5 is a protocol diagram of a second, eager rendezvous transmission mode in which message transmission is initiated using a packet having both control information and a data portion of the message, with any remaining data portions of the message being transmitted following an acknowledgement from the destination node, in an analogous art for the purpose of Arndt (as suggested in title and abstract).

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Arndt teachings on managing server farms, to add the teachings of Franke.

The modifications would have been obvious because one of ordinary skill in the art would have been motivated for a method, system, and associated program code and data structures, protocols and buffering for facilitating the efficient transmission of messages from a source node to a destination node in a message processing system which prevent the performance degradation associated with packet retransmission after timeouts.

Together Arndt and Franke disclosed all limitations of **Claims 1-20,** are rejected under 35 U.S.C. 103(a).

As to Claims 1, 11, 17 and 20, *Arndt teaches method, compute node and computer system for communicating data from a first compute node of a computer system comprising multiple compute nodes interconnected by an inter-node*

communication network to a second one of the multiple compute nodes, the method comprising (as stated in par. 0029, lines 1-13, par. 0030, lines 1-5, par. 0033, lines 1-12, FIG. 1 is a diagram of a distributed computer system, takes the form of a system area network (SAN) 100 having end nodes, switches, routers, and links interconnecting these components, and can be implemented on computer systems of numerous other types and configurations. Computer systems can range from a small server with one processor and a few input/output (I/O) adapters to massively parallel supercomputer systems with hundreds or thousands of processors and thousands of I/O adapters. SAN 100 is a high-bandwidth, low-latency network interconnecting nodes within the distributed computer system. A node is any component attached to one or more links of a network and forming the origin and/or destination of messages within the network, where a message is an application-defined unit of data exchange, which is a primitive unit of communication between cooperating processes. SAN 100 contains the communications and management infrastructure supporting both I/O and Interprocessor communications (IPC) within a distributed computer system. The SAN 100 shown in FIG. 1 includes a switched communications fabric 116, which allows many devices to concurrently transfer data with high-bandwidth and low latency in a secure, remotely managed environment. End nodes can communicate over multiple ports and utilize multiple paths through the SAN fabric. The multiple ports and paths through the SAN shown in FIG. 1 can be employed for fault tolerance and increased bandwidth data transfers):

placing the data on a full-duplex packetized interconnect directly connecting a CPU of the first compute node to a network interface connected to the inter-node communication network (as stated in par. 0034, lines 1-5, par. 0037, lines 1-15, par. 0038, lines 1-3, The SAN 100 in FIG. 1 includes switch 112, switch 114, switch 146, and router 117. A switch is a device that connects multiple links together and allows routing of packets from one link to another link within a subnet using a small header Destination Local Identifier (DLID) field. A link is a full duplex channel between any two network fabric elements, such as end nodes, switches, or routers. In SAN 100 as illustrated in FIG. 1, host processor node 102, host processor node 104, and I/O chassis 108 include at least one channel adapter (CA) to interface to SAN 100. In one embodiment, each channel adapter is an endpoint that implements the channel adapter interface in sufficient detail to source or sink packets transmitted on SAN fabric 116. Host processor node 102 contains channel adapters in the form of host channel adapter 118 and host channel adapter 120. Host processor node 104 contains host channel adapter 122 and host channel adapter 124. Host processor node 102 also includes central processing units 126-130 and a memory 132 interconnected by bus system 134. Host processor node 104 similarly includes central processing units 136-140 and a memory 142 interconnected by a bus system 144. Host channel adapters 118 and 120 provide a connection to switch 112 while host channel adapters 122 and 124 provide a connection to switches 112 and 114);

receiving the data at the network interface (as stated in par. 0039, lines 1-8, par. 0040, lines 1-16, par. 0042, lines 1-11, The host channel adapter hardware offloads

much of central processing unit and I/O adapter communication overhead. This hardware implementation of the host channel adapter also permits multiple concurrent communications over a switched network without the traditional overhead associated with communicating protocols. As in FIG. 1, router 117 is coupled to wide area network (WAN) and/or local area network (LAN) connections to other hosts or other routers. The I/O chassis 108 in FIG. 1 includes an I/O switch 146 and multiple I/O modules 148-156. I/O adapters also include a switch in the I/O adapter backplane to couple the adapter cards to the SAN fabric. These modules contain target channel adapters 158-166. The I/O modules take the form of adapter cards. SAN 100 handles data communications for I/O and Interprocessor communications. SAN 100 supports high-bandwidth and scalability required for I/O and also supports the extremely low latency and low CPU overhead required for Interprocessor communications. User clients can bypass the operating system kernel process and directly access network communication hardware, such as host channel adapters, which enable efficient message passing protocols. SAN 100 is suited to current computing models and is a building block for new forms of I/O and computer cluster communication);

and, transmitting the data to a network interface of the second compute node by way of the inter-node communication network (as stated in par. 0042, lines 11-17, par. 0043, lines 1-2, par. 0044, lines 1-10, Further, SAN 100 in FIG. 1 allows I/O adapter nodes to communicate among themselves or communicate with any or all of the processor nodes in distributed computer system. With an I/O adapter attached to the SAN 100, the resulting I/O adapter node has substantially the same communication

capability as any host processor node in SAN 100. SAN 100 shown in FIG. 1 supports channel semantics and memory semantics. In channel semantics, the destination process pre-allocates where to place the transmitted data. In memory semantics, a source process directly reads or writes the virtual address space of a remote node destination process. The remote destination process need only communicate the location of a buffer for data, and does not need to be involved in the transfer of any data. Thus, in memory semantics, a source process sends a data packet containing the destination buffer memory address of the destination process. In memory semantics, the destination process previously grants permission for the source process to access its memory. Channel semantics and memory semantics are typically both necessary for I/O and Interprocessor communications).

As to Claims 2 and 12, *Arndt teaches method and compute node according to claims 1 and 11, wherein the network interface and the CPU are the only devices configured to place data on the packetized interconnect* (as stated in par. 0047, lines 5-15, par. 0048, lines 1-13, par. 0053, lines 1-16, host processor node 200 shown in FIG. 2 includes a set of consumers 202-208, which are processes executing on host processor node 200. Host processor node 200 also includes channel adapter 210 and channel adapter 212. Channel adapter 210 contains ports 214 and 216 while channel adapter 212 contains ports 218 and 220. Each port connects to a link. The ports can connect to one SAN subnet or multiple SAN subnets, such as SAN 100 in FIG. 1. Channel adapters take the form of host channel adapters. Consumers 202-208 transfer

messages to the SAN via the verbs interface 222 and message and data service 224. Message and data service 224 provides an interface to consumers 202-208 to process messages and other data. Channel adapters, switches, and routers employ multiple virtual lanes within a single physical link. As illustrated in FIGS. 3A, 3B, and 3C, physical ports connect end nodes, switches, and routers to a subnet. Packets injected into the SAN fabric follow one or more virtual lanes from the packet's source to the packet's destination. The virtual lane that is selected is mapped from a service level associated with the packet. At any one time, only one virtual lane makes progress on a given physical link. Virtual lanes provide a technique for applying link level flow control to one virtual lane without affecting the other virtual lanes).

As to Claims 3 and 18, *Arndt teaches method and compute node according to claims 1 and 11, comprising transmitting the data from the network interface to the second computer node by way of a full-duplex communication link of the inter-node communication network* (as stated in par. 0042, lines 1-6, par. 0035, lines 1-6, par. 0086, lines 4-9, par. 0087, lines 3-11, par. 0088, lines 1-10, SAN 100 handles data communications for I/O and Interprocessor communications. SAN 100 supports high-bandwidth and scalability required for I/O and also supports the extremely low latency and low CPU overhead required for Interprocessor communications. A link is a full duplex channel between any two network fabric elements, such as end nodes, switches, or routers. Data packets are routed through the SAN fabric, and for reliable transfer services, are acknowledged by the final destination endnode. A host name provides a

logical identification for a host node, such as a host processor node or I/O adapter node. The host name identifies the endpoint for messages such that messages are destined for processes residing on an end node specified by the host name. Thus, there is one host name per node, but a node can have multiple channel adapters, CAs. A single IEEE assigned 64-bit identifier (EUI-64) 902 is assigned to each component. A component can be a switch, router, or channel adapter CA. One or more globally unique ID (GUID) identifiers 904 are assigned per channel adapter CA port 906. Multiple GUIDs (a.k.a. IP addresses) can be used for several reasons, for example, different IP addresses identify different partitions or services on an end node, different IP addresses are used to specify different Quality of Service (QoS) attributes and, different IP addresses identify different paths through intra-subnet routes.

As to Claims 4 and 16, *Arndt teaches method and compute node according to claims 3 and 11, comprising passing the data through a buffer at the network interface before transmitting the data* (as stated in par. 0049, lines 6-18, Buffering of data to host channel adapter ports 312A-316A is channeled through virtual lanes (VL) 318A-334A where each VL has its own flow control. Subnet manager configures channel adapters with the local addresses for each physical port, i.e., the port's LID. Subnet manager agent (SMA) 336A is the entity that communicates with the subnet manager for the purpose of configuring the channel adapter. Memory translation and protection (MTP) 338A is a mechanism that translates virtual addresses to physical addresses and

validates access rights. Direct memory access (DMA) 340A provides for direct memory access operations using memory 342A with respect to queue pairs 302A-310A).

As to Claims 5 and 19, *Arndt teaches method and compute node according to claims 1 and 11, comprising, at the network interface, determining a size of the data and, based upon the size of the data, selecting among two or more protocols for transmitting the data* (as stated in par. 0058, lines 1-7, par. 0060, lines 1-11, par. 0067, lines 1-6, Send work queue 402 contains work queue elements (WQEs) 422-428, describing data to be transmitted on the SAN fabric. Receive work queue 400 contains work queue elements (WQEs) 416-420, describing where to place incoming channel semantic data from the SAN fabric. A work queue element is processed by hardware 408 in the host channel adapter. A send work request is a channel semantic operation to push a set of local data segments to the data segments referenced by a remote node's receive work queue element. Work queue element 428 contains references to data segment 4 438, data segment 5 440, and data segment 6 442. Each of the send work request's data segments contains a virtually contiguous memory space. The virtual addresses used to reference the local data segments are in the address context of the process that created the local queue pair. When a queue pair is created, the queue pair is set to provide a selected type of transport service. A distributed computer system implementing the present invention supports four types of transport protocol services: reliable connection, unreliable connection, reliable datagram, and unreliable datagram connection service).

As to Claims 8, 10 and 15, *Arndt teaches method and compute node according to claims 1 and 11, wherein the data comprises a raw ethertype datagram and transmitting the data comprises encapsulating the raw ethertype datagram within one or more link layer packet headers* (as stated in par. 0104, lines 1-12, par. 0109, lines 1-12, An outgoing message is split into one or more data packets. The channel adapter hardware adds a transport header and a network header to each packet. The transport header includes sequence numbers and other transport information. The network header includes routing information, such as the destination IP address and other network routing information. The link header contains the Destination Local Identifier (DLID) or other local routing information. The appropriate link header is always added to the packet. The appropriate global network header is added to a given packet it the destination end node resides on a remote subnet. Consumers 1103 and 1105 represent applications or processes that employ the other layers for communicating between end nodes. Transport layer 1104 provides end-to-end message movement. In one embodiment, the transport layer provides four types of transport services: reliable connection service; reliable datagram service; unreliable datagram service; and raw datagram service. Network layer 1106 performs packet routing through a subnet or multiple subnets to destination end nodes. Link layer 1108 performs flow-controlled, error checked, and prioritized packet delivery across links).

As to Claim 9, *Arndt teaches method according to claim 8 wherein the link layer packet headers comprise InfiniBand.TM. link layer packet headers* (as stated in par. 0113, lines 1-12, SAN environment described above with regard to FIGS. 1-12 satisfies the InfiniBand requirement of a well-known QP0 communication channel being provided for each logical port on a logical HCA and also for each logical switch. Rather than including separate physical resources for each of these low-utilization communications channels, a single physical QP0 and its associated firmware are provided for each physical port. SAN environment provides mechanisms for routing and processing this QP0 traffic on behalf of multiple logical ports when there is only a single QP0 associated with the physical port).

As to Claim 6, *Together Arndt and Franke teach method according to claim 5 wherein the two or more protocols comprise an eager protocol and a rendezvous protocol* (as stated by Frank in Abstract lines 1-12, An "eager" rendezvous transmission mode is disclosed in which early arrival buffering is provided at message destination nodes for a predetermined amount of data for each of a predetermined number of incoming messages. Relying on the presence of the early arrival buffering at a message destination node, a message source node can send a corresponding amount of message data to the destination node along with control information in an initial transmission).

As to Claim 7, *Together Arndt and Franke teach method according to claim 6 comprising, upon selecting the rendezvous protocol, automatically generating a Ready To Send message at the network interface of the first compute node* (as stated by Frank in Abstract lines 12-24, Any remaining message data is sent only upon receipt by the source node of an acknowledgement from the destination node indicating that the destination node is prepared to receive any remaining data. In an enhanced embodiment, the source node alternates between rendezvous transmission modes as a function of the amount of free space in the early arrival buffering at the destination node, as indicated by the number of outstanding initial transmissions for which acknowledgements have not yet been received. Different transmission modes for different destination nodes can be employed at a source node, depending on the amount of early arrival buffering currently available in each respective destination node).

As to Claim 13, *Together Arndt and Franke teach compute node according to claim 11 comprising a memory and a facility configured to allocate eager protocol buffers in the memory and to automatically signal to one or more other compute nodes that the eager protocol buffers have been allocated* (as stated by Frank , in col. 6, lines 30-49, col. 11, lines 19-26, The length "N" of the first data portion of the message transmitted is a predetermined number which should correspond to the size "N" of the early arrival buffer slot pre-allocated at the destination node for each message of a number "Q" of messages. If "N" is large enough so that the destination node receives the control information in the initial transmission and returns the rendezvous

acknowledgement before all "N" bytes have been sent by the source node, then the source node does not experience any interruption in the data transmission and the destination, likewise, does not see any interruption in the data received. This eager rendezvous transmission mode does not require a large amount of buffering at the destination for these initial transmissions since each such early arrival transmission only brings with it, at most, "N" bytes of the data portion of the message. The early arrival buffering may be a "flat" buffer, possibly from a buffer pool, or can be implemented using pointers and/or linked lists. The principles of transmission mode 300 of FIG. 6 can be applied across a system having multiple source and destination nodes. Each source node maintains the count C of each of the unacknowledged initial eager rendezvous transmissions sent to each respective destination node, and alternates between the rendezvous transmission modes 100 and 200 on a per destination node basis, depending upon the count C for each respective destination node).

As to Claim 14, *Together Arndt and Franke teach compute node according to claim 13 comprising a facility configured to automatically associate memory protection keys with the eager protocol buffers and a facility configured to verify memory protection keys in incoming eager protocol messages before writing the incoming eager protocol messages to the eager protocol buffers.* (as stated by Arndt, in par. 0046, lines 1-5, par. 0063, lines 1-5, par. 0064, lines 1-10, the distributed computer system shown in FIG. 1 performs operations that employ virtual addresses and virtual memory protection mechanisms to ensure correct and proper access to all memory. A RDMA Write work

queue element provides a memory semantic operation to write a virtually contiguous memory space on a remote node. The RDMA Write work queue element contains a scatter list of local virtually contiguous memory spaces and the virtual address of the remote memory space into which the local memory spaces are written. A RDMA FetchOp work queue element provides a memory semantic operation to perform an atomic operation on a remote word. The RDMA FetchOp work queue element is a combined RDMA Read, Modify, and RDMA Write operation. The RDMA FetchOp work queue element can support several read-modify-write operations, such as Compare and Swap if equal. A bind (unbind) remote access key (R_Key) work queue element provides a command to the host channel adapter hardware to modify (destroy) a memory window by associating (disassociating) the memory window to a memory region. The R_Key is part of each RDMA access and is used to validate that the remote process has permitted access to the buffer).

Action Final

4. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Muktesh G. Gupta whose telephone number is 571-270-5011. The examiner can normally be reached on Monday-Friday, 8:00 a.m. -5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William C. Vaughn can be reached on 571-272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MG

/William C. Vaughn, Jr./

Supervisory Patent Examiner, Art Unit 2444